

非同期 P2P 通信ミドルウェア—通信制御の研究—

†澤口宗和 †伊東正起 †大谷真 †松原裕人

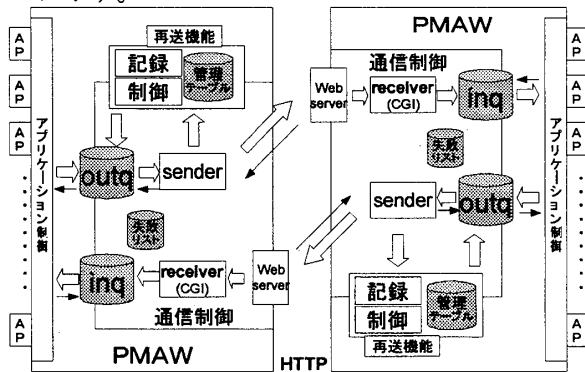
湘南工科大学 情報工学科

1. はじめに

今や Web は全世界に普及し、ブラウザ表示だけでなく、いわゆる Web サービスのように異なるサイトのアプリケーション(AP)を接続するためのインフラとなった。一方、本格的な企業間取引の場合、現状の RPC をベースとした Web サービスでは十分でなく、非同期メッセージング型の接続形態が必要とされている。このため現在、Web 上で非同期対等(P2P)型の通信を行うミドルウェア(PMAW)を開発している。本論文はPMAWの通信制御部について開発方針とプロトタイプ実装結果について述べる。

2. PMAW 通信制御部の開発方針

store and forward 方式のキューをデータベース(以下 DB)を用いて構成し、非同期通信を実現する。双方に Web サーバを配置し、通信プロトコルには通常の Web プロトコル(HTTP, XML)を用いる。通信制御を中心とした PMAW の構成を図 1 に示す。



inq:受信キュー, outq:送信キュー, AP:ユーザ AP  
 sender:送信用プログラム  
 receiver:受信用プログラム  
 ⇨ :ユーザデータの流れ, →:主な処理の流れ

図 1 非同期 P2P 通信ミドルウェアの構成

PMAW では一案件の通信を長期かつ一意に保つ LS(Long Session)という概念を用いる[1]。AP は API を用いてデータを送受信する。双方で LS を確立しセッションの ID(lsid)を作成後、lsid を用いてデータ交換を行う。データ交換が

終了したら LS を破棄する。AP がデータを送信する場合 DB を介して sender から送信、すぐに receiver からのレスポンスを受けとり送信側 AP に返す。受信側 AP はデータを非同期に受信する。再送機能や失敗リストは後述する Exactry Once と Sequence 保障を実現するための機能である。

3. 通信制御部の開発

通信制御部における最大の課題はメッセージングサービスの信頼性 Exactly Once と Sequence 保障の実現にある。Exactly Once とは必ず一つのデータが送られると言う意味でありデータの重複や抜けは許されない。Sequence 保障とはデータの順序保障の意味である。ユーザ AP がデータを送信した順番どおり相手 AP に届かなければならない。

AP 制御部と通信制御部は DB のキューを介してデータの受け渡しを行う。通信プロトコルについては、OASIS 標準[2]を簡略化した形とした。

4. 実現方法

4. 1 通信プロトコル

SOAP フレームワーク標準により XML のデータを HTTP でやり取りする(表 1 参照)。受信側は Web サーバから cgi を呼び出し標準出力でレスポンスを返す(表 2 参照)。送信側はレスポンスの受信結果を outq に書き込む。一方受信側は inq にデータを格納し、AP 制御に処理を渡す。

表 1 SOAP 送信データ

SOAP ヘッダ部			
lsid	セッションの ID		
s_ua	自ユーザ名	r_ua	相手ユーザ名
s_url	自 URL	r_url	相手 URL
apc_id	AP 制御 id	message_num	メッセージ通番

SOAP ボディ部			
s_op	自 op 名	r_op	相手 op 名
message_data	ユーザデータ	fault_data*1	送信失敗通番

\*1:データの順序性の項で後述

表 2 SOAP レスポンスデータ

SOAP ヘッダ部			
lsid	セッションの ID	message_num	メッセージ通番

SOAP ボディ部			
status	送信結果	fault_data*1	仮登録通番
er_num	エラー番号	place	エラー箇所
re_trans_param	cgi での再送情報 *2		

\*1:データの順序性の項で後述

\*2:データの再送の項で後述

Asynchronous P2P Communication Middleware  
 -Study on Communication Manager-

† Hirokazu Sawaguchi, Masaki Itou, Hiroto Matubara, Makoto Oya - Shonan Institute of Tecnology

#### 4. 2データの再送

図2に再送機能の流れを示す。sender の送信中にエラーが起きた場合、エラーの発生箇所と種類から待ち時間を決定し再送信をする。HTTP や SQL、ソケットに規定されるコードに対応したレベルを決定する。レベルは 0~6 までありそれぞれ再送を行う待ち時間の間隔が異なる。再送回数には上限を設け、10 回までとした。

再送データは outq にて再送待ち状態となり、エラー箇所とコードが管理テーブルに書き込まれ、記録プログラムが待ち時間を決定し次回送信予定時間が記録される。制御プログラムによって予定時間を満了したデータが outq にて送信待ち状態となる。送信が成功するか送信諦めとなると管理テーブルのデータは削除される。

受信側 cgi や inq への書き込みの際の sql エラーは相手の都合次第で再送の待ち時間が変化する。よって lsid の生成時にお互いの再送の都合を送り合う仕組みになっている(表 2\*2)。

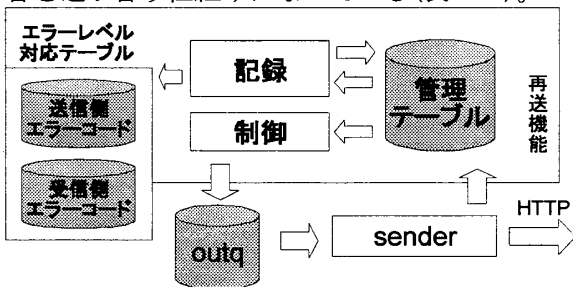


図2 再送機能

#### 4. 3データの順序性

##### 4. 3. 1メッセージ通番

lsid 毎に outq からデータを取り出した時点でメッセージ通番を付加する。inq に届くまで、すなわち通信路上ではデータの追い抜きを認めている。データを受信した cgi は inq にデータを登録した後、登録した lsid の連番の最大値を max\_num テーブルに保管する。これによりこの最大値より小さい値はシーケンスが保障されたデータとみなすことが出来る。データを AP 制御に渡す際には AP 制御が通信制御のプログラムを呼び出す。プログラムは inq から max\_num テーブルに保管された最大値より小さい値でまだ AP 制御に渡していないデータを取り出し、AP 制御に渡す。

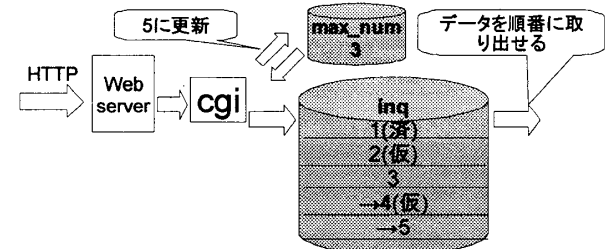
##### 4. 3. 2仮データ

データの中には送信、再送に失敗し相手に送信できないデータが発生する恐れがある。メッセージ通番で順序性を判定する為、このような場合は送信に失敗したデータのメッセージ通番を補完する必要がある。よって失敗リストテ

ブルという失敗したデータのリストを送信側に用意する。送信側は送信の際に失敗リストテーブル内の自分と同じ lsid のメッセージ通番複数と一緒に送信する(表 1\*1)。受信側はそのメッセージ通番を用いて仮のデータ作成し、inq に登録する。送信側はレスポンスから失敗データの登録が出来たかどうかを確認し、成功ならば失敗データのリストに送信済の旨を伝える(表 2\*1)。これにより見かけ上 inq には通番が穴抜けなく入っているかに見えるのでシーケンスにデータを取り出すことが可能である。

##### 4. 3. 3シーケンスの保障の流れ

例えば図3のように 1~5 番までのデータを送信する時、2,4 番のデータが送信失敗となった場合、3 番データが 2 番の、5 番が 4 番の通番を添えて送信され、inq に仮データとして登録される。図3は 4,5 のデータを inq に挿入し、max\_num テーブルを更新している様子である。



番号:メッセージ通番, (仮):仮データ  
(済):AP に送信済, →: 挿入されたデータ

図3 シーケンス保障の処理方法

#### 5. まとめ

本論文で述べた方式を使い通信制御部のプロトタイプを開発した。開発環境には PHP5. 2. 3, Apache2. 2. 4, PostgreSQL8. 2 を用いた。同時に開発が行われた AP 制御と連動テストを行い非同期メッセージング、Exactly Once、Sequence 保障等の目標機能が正しく動作することを確認した。これによって本論文の方式で述べた方式の妥当性が検証できた。

#### 6. 参考文献

1. 伊東 他, 非同期通信ミドルウェア-アプリケーション制御部の研究-, 情報処理第 70 回全国大会, 2008
2. OASIS, " ebxml Messageing Services 3.0" , OASIS Standard, 2007  
[http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms\\_core-3.0-spec-os.html](http://docs.oasis-open.org/ebxml-msg/ebms/v3.0/core/os/ebms_core-3.0-spec-os.html)